

Exercise: Learning About Each Other's Projects

Take five minutes to prepare one-minute summary of your project to share with the class members.

Use the terms on the "Ontology" handout to describe your project.

Exercise: What Should Remain on the Shelf in 20 Years

What matters about your project?

What research questions is it designed to enable and answer? What components/data/text/images/metadata/functionalities make it possible to answer those questions?

What resources are most often used by you? By your users?

What parts of your project do other people cite and link to?

What will users in twenty years find useful about your project? Conversely, what won't matter at all?

Exercise: How is your data managed and stored? Do you need to make changes there?

Where is your data stored?

How many copies exist?

How many distinct and independent places are they in? (Think about geography -- earthquake zones, national boundaries -- and dependencies. A place is not independent if it depends on storage in another place.)

If you are hit by a bus tomorrow, who will take on responsibility for your data? How will they know what to do? (Passwords, access, etc)

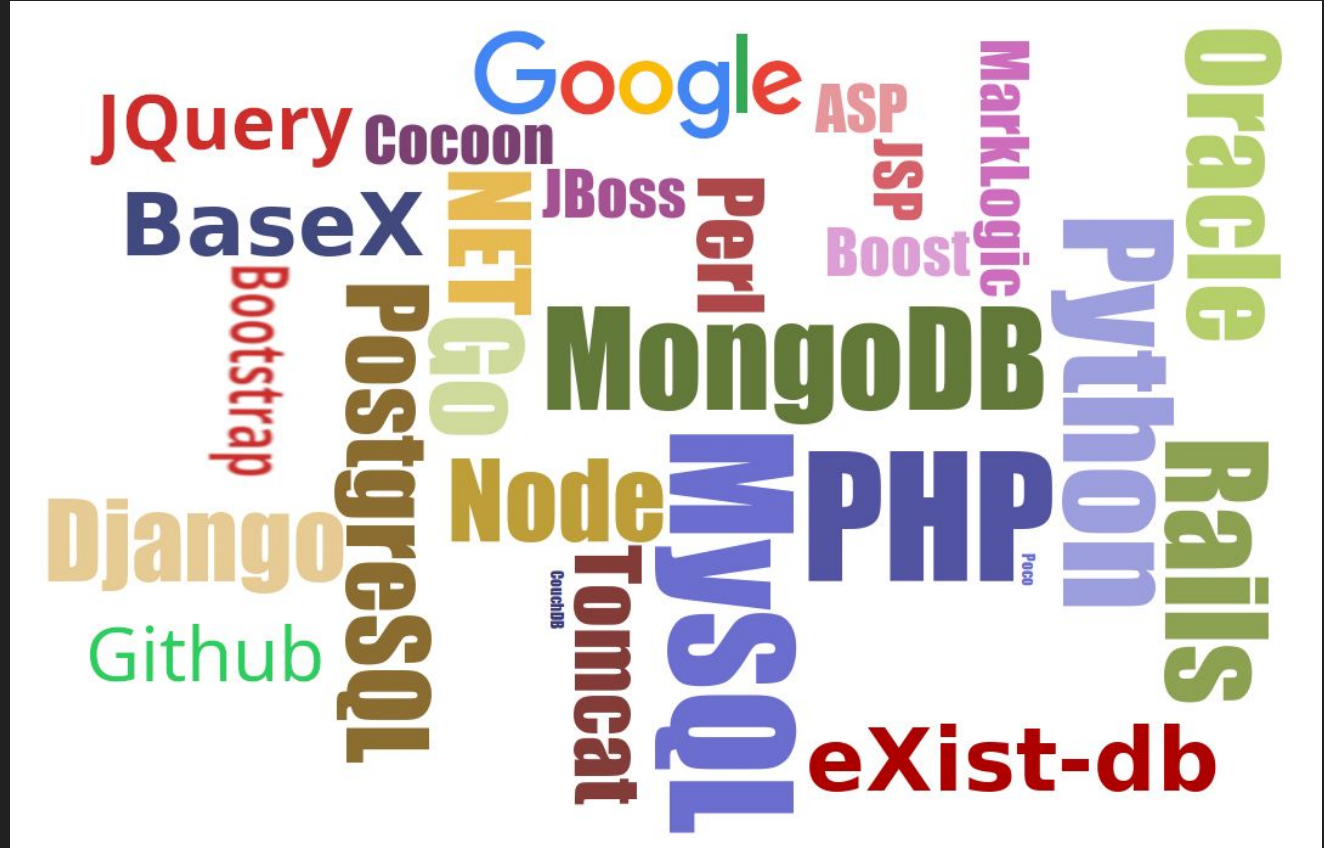
Is your data in a format that can be read by humans without proprietary technologies?

Exercise: List your project's products.

Products are things like: website, print documents, image collections, XML files, statistical analyses, epub, PDFs, GeoJSON, RDF.

(Your list should be entirely comprised of nouns.)

Exercise: Enumerate the dependencies of your products.



Discussion: If our dependencies fail, which of our products will still be available (we think)?

Exercise: List the issues from your Archive.org run-through that need to be addressed.

Broken links

Missing pages, images, Javascript libraries

Broken maps

Things that don't display the way you designed them.

404 errors

Search functions that don't return answers

Functionalities that don't work (e.g., pop-ups, mouse-overs)

Exercise: Enumerate the aspects of *your* project that could be tracked by automated diagnostics.

Coherence and consistency

Completeness

Progress and predictions

Exercise: Write some diagnostics code!

Janelle will help you nail down what you need diagnostics to do.

Joey and Martin will help you write some code..

Exercise: What will go into your first/next release?

What content will you release?

What content will you feature (if any)?

What corrections will you make? What gaps will you fill?

What technological affordances will you launch?

Exercise: Refining your diagnostics for release tracking.

Janelle will help you nail down what you need Diagnostics to do.

Joey and Martin will help you refine your code.

Exercise: Is your documentation adequate?

Where is your documentation? What language is it written in?

Can other people find it? Read it? (Note that code is not self-documenting for most users.)

Have you documented the following?

- How you prepared your data/texts: What decisions you made along the way
- The scope of your project: What you excluded, what you included, and why
- How you encoded it (if your project entails encoding)
- The output languages
- Programming -- if you had to replace your programmer, is there enough documentation for someone else to step in?

Discussion: How practical is a static version of your site?

This final discussion is your chance to make your take-home to-do list. All sites can be staticized. What do you still have to do to make your site staticizable?

Spend five minutes making your list. Then we'll share and discuss lists. You will likely add to your own list as you hear from other people.